

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-7359

(P2002-7359A)

(43) 公開日 平成14年1月11日 (2002.1.11)

(51) Int.Cl. ⁷	識別記号	F I	テ-マ-ト* (参考)
G 0 6 F 15/16	6 1 0	G 0 6 F 15/16	6 1 0 A 5 B 0 4 5
15/167		15/167	C 5 B 0 5 7
G 0 6 T 1/20		G 0 6 T 1/20	B

審査請求 未請求 請求項の数 6 O L (全 10 頁)

(21) 出願番号 特願2000-186226(P2000-186226)

(22) 出願日 平成12年6月21日(2000.6.21)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 倉田 徹

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(74) 代理人 100094053

弁理士 佐藤 隆久

Fターム(参考) 5B045 D003 D011 G014

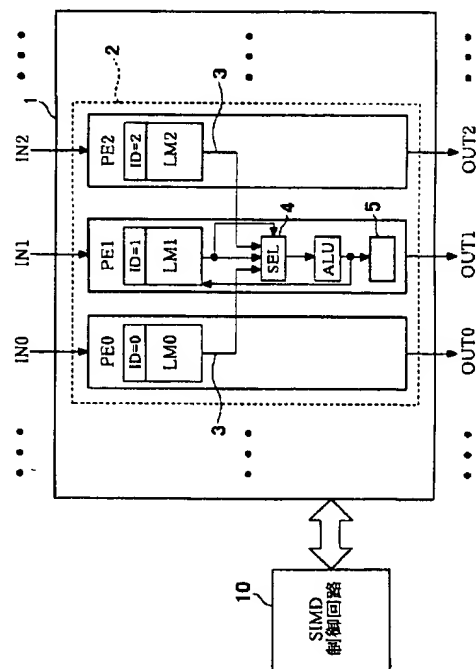
5B057 C008 C004 C009 C011

(54) 【発明の名称】 SIMD制御並列処理方法および装置

(57) 【要約】

【課題】処理に実質的に寄与していない要素プロセッサ (PE) を活用して、メモリ容量および処理速度を上げる。

【解決手段】1次元的に配列された複数のPEを同一の指令により制御するSIMD制御において、たとえばPE0、PE1、PE2からなる複数のグループ2ごとに同じ配列で繰り返し割り当てられ、かつ、各グループ2内で任意のPEを特定可能なIDコードを各PEごとに付与する。そして、PE0~PE2にデータを入力し同一の処理を実行させ、グループ2内の特定のPE1の処理の途中結果を、特定の識別コード(ID=1)を基準に指定した他のPE0、PE2に記憶させ、また記憶した途中結果を(ID=1)を基準にして読み出して後続の処理に用いる。処理の結果を(ID=1)を基準に選択して出力させる。



【特許請求の範囲】

【請求項1】1次元的に配列された複数の要素プロセッサを単一の命令により制御するSIMD制御並列処理方法であって、

所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、

上記複数の要素プロセッサにデータを入力して、同一の処理を実行させ、

上記グループ内の特定の要素プロセッサにおいて、上記処理の途中結果を、特定の識別コードを基準に指定した他の要素プロセッサに記憶させ、当該記憶した途中結果を、上記特定の識別コードを基準とした指定により読み出して後続の処理に用いるSIMD制御並列処理方法。

【請求項2】上記処理の結果を、特定の識別コードを基準とした指定により選択して出力させる請求項1記載のSIMD制御並列処理方法。

【請求項3】1次元的に配列された複数の要素プロセッサを単一の命令により制御するSIMD制御並列処理方法であって、

所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、

上記複数の要素プロセッサにデータとパラメータを入力して、同一の処理を実行させ、

上記要素プロセッサの処理結果の幾つかを、識別コードを基準に指定して統合処理し、

上記統合処理の結果を、識別コードを基準とした指定により選択して出力させるSIMD制御並列処理方法。

【請求項4】情報を記憶するメモリ部、メモリ部に記憶された情報を基に処理を実行する処理部をそれぞれに含んで1次元的に配列され、隣接相互間でデータ通信が可能な複数の要素プロセッサと、

上記複数の要素プロセッサを単一の命令によりSIMD制御するSIMD制御回路とを有し、

上記SIMD制御に、所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、上記複数の要素プロセッサにデータを入力して、同一の処理を実行させ、上記グループ内の特定の要素プロセッサにおいて、上記処理の途中結果を、特定の識別コードを基準に指定した他の要素プロセッサに記憶させ、当該記憶した途中結果を、上記特定の識別コードを基準とした指定により読み出して後続の処理に用いる制御を含むSIMD制御並列処理装置。

【請求項5】上記SIMD制御に、さらに、上記処理の結果を上記特定の識別コードを基準とした指定により選

択して出力させる制御を含む請求項4記載のSIMD制御並列処理装置。

【請求項6】情報を記憶するメモリ部、メモリ部に記憶された情報を基に処理を実行する処理部をそれぞれに含んで1次元的に配列され、隣接相互間でデータ通信が可能な複数の要素プロセッサと、

上記複数の要素プロセッサを単一の命令によりSIMD制御するSIMD制御回路とを有し、

上記SIMD制御に、所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、上記複数の要素プロセッサにデータとパラメータを入力して、同一の処理を実行させ、上記要素プロセッサの処理結果の幾つかを、識別コードを基準に指定して統合処理し、上記統合処理の結果を、識別コードを基準とした指定により選択して出力させる制御を含むSIMD制御並列処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、数多くのPE(Processor Element)に対して単一の命令で多並列処理を実行するSIMD(Single Instruction Multiple Data)制御方式の並列処理装置およびその制御方法に関する。

【0002】

【従来の技術】SIMD制御の多並列プロセッサでは、複数の要素プロセッサPEを1次元に配置し、当該複数の要素プロセッサ(PE)を単一の命令により制御するものである。従来、このSIMD制御の多並列プロセッサでは、同一のデータを割り当てることはなく、各要素プロセッサに対してデータが常に1対1もしくは1対多になっていた。また、要素プロセッサ間通信を行う機能も有するが、この機能はあくまで、一つの処理に他の要素プロセッサの結果を利用する場合に用いるために設けられていた。

【0003】

【発明が解決しようとする課題】従来のSIMD制御の多並列プロセッサにおいて、しばしばローカルメモリ(LM)の不足や、処理ステップ数の不足が問題となっていた。また、SIMD制御プロセッサの構造上要素プロセッサ数が固定されているのに対し、データ・ストリームの長さが必ずしも要素プロセッサ数と一致しているとは限らないため、実際には使用されることのない要素プロセッサが数多く存在する場合があり、これらの要素プロセッサは無駄となっていた。また、データ・ストリームの中で所定の異なる処理を周期的に行う処理も多い。SIMD制御では、所定数おきに要素プロセッサの一つが有効な処理中のとき、他の要素プロセッサも同じ手順の処理を行うが、周期的な処理では、他の要素プロセッサの処理結果は出力されない。このため、他の要素

プロセッサの処理は、有効な処理に何ら寄与しておらず、この点でも無駄が多かった。

【0004】本発明の目的は、このような要素プロセッサが無駄になるデータ・ストリームに対して、要素プロセッサを有効利用したうえでローカルメモリを実質的に複数倍に増加させる処理を、ハードウェアの構成をそのままに実現することにある。また、本発明の他の目的は、フィルタリング処理等、パラメータが異なるだけで全く同じような処理を繰り返すような処理の場合に、従来では出力データの算出に全く寄与していなかった要素プロセッサを有効利用し、並列処理数を実質的に複数倍に増加させた効率が低い処理を、ハードウェアの構成をそのままに実現することにある。

【0005】

【課題を解決するための手段】本発明の第1の観点に係るSIMD制御並列処理方法は、1次元的に配列された複数の要素プロセッサを単一の命令により制御するSIMD制御並列処理方法であって、所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、上記複数の要素プロセッサにデータを入力して、同一の処理を実行させ、上記グループ内の特定の要素プロセッサにおいて、上記処理の途中結果を、特定の識別コードを基準に指定した他の要素プロセッサに記憶させ、当該記憶した途中結果を、上記特定の識別コードを基準とした指定により読み出して後続の処理に用いる。好適に、上記処理の結果を、特定の識別コードを基準とした指定により選択して出力させる。

【0006】本発明の第2の観点に係るSIMD制御並列処理方法は、1次元的に配列された複数の要素プロセッサを単一の命令により制御するSIMD制御並列処理方法であって、所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、上記複数の要素プロセッサにデータとパラメータを入力して、同一の処理を実行させ、上記要素プロセッサの処理結果の幾つかを、識別コードを基準に指定して統合処理し、上記統合処理の結果を、識別コードを基準とした指定により選択して出力させる。

【0007】本発明の第3の観点に係るSIMD制御並列処理装置は、情報を記憶するメモリ部、メモリ部に記憶された情報を基に処理を実行する処理部をそれぞれに含んで1次元的に配列され、隣接相互間でデータ通信が可能な複数の要素プロセッサと、上記複数の要素プロセッサを単一の命令によりSIMD制御するSIMD制御回路とを有し、上記SIMD制御に、所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサ

を特定可能な識別コードを各要素プロセッサごとに付与し、上記複数の要素プロセッサにデータを入力して、同一の処理を実行させ、上記グループ内の特定の要素プロセッサにおいて、上記処理の途中結果を、特定の識別コードを基準に指定した他の要素プロセッサに記憶させ、当該記憶した途中結果を、上記特定の識別コードを基準とした指定により読み出して後続の処理に用いる制御を含む。好適に、上記SIMD制御に、さらに、上記処理の結果を上記特定の識別コードを基準とした指定により選択して出力させる制御を含む。

【0008】本発明の第4の観点に係るSIMD制御並列処理装置は、情報を記憶するメモリ部、メモリ部に記憶された情報を基に処理を実行する処理部をそれぞれに含んで1次元的に配列され、隣接相互間でデータ通信が可能な複数の要素プロセッサと、上記複数の要素プロセッサを単一の命令によりSIMD制御するSIMD制御回路とを有し、上記SIMD制御に、所定数の要素プロセッサからなるグループごとに同じ配列で繰り返し割り当てられ、かつ、各グループ内で任意の要素プロセッサを特定可能な識別コードを各要素プロセッサごとに付与し、上記複数の要素プロセッサにデータとパラメータを入力して、同一の処理を実行させ、上記要素プロセッサの処理結果の幾つかを、識別コードを基準に指定して統合処理し、上記統合処理の結果を、識別コードを基準とした指定により選択して出力させる制御を含む。

【0009】このような本発明に係るSIMD制御並列処理方法および装置では、認識コードの繰り返し配列付与により、要素プロセッサ群を所定数の要素プロセッサからなるグループに分け、このグループ内の複数の要素プロセッサを1つの仮想的な要素プロセッサとして動作させる。

【0010】具体的に、上記第1および第3の観点に係るSIMD制御並列処理方法および装置では、複数の要素プロセッサのメモリ部を1つのデータに対して仮想的に割り当てて使用する。このメモリ部への書き込みおよび読み出しでは、認識コードを基にメモリ部の指定を行う。認識コードを基にしたメモリ指定では、メモリ部への書き込みおよび読み出しを、基準となる要素プロセッサのみ正常にできる。他の要素プロセッサでは、書き込み先が指定されなかったり、データを別の場所から読み出すため、処理結果が誤ったものとなる。そして、基準となる要素プロセッサのみから正常な出力データが得られる。このような制御では、要素プロセッサ・ブロック内の要素プロセッサ数を n とした場合、並列処理装置全体として一度に処理できるデータ数は $1/n$ になるものの、1つのデータに対して使用できるメモリ容量を実質的には n 倍にすることができる。このような仮想的なメモリ拡張手法は、間欠的にしか出力データは要らないが、処理時に多量のメモリ空間を必要とする場合に適している。また、SIMD制御回路のプログラム制御で行

えるため、従来のSIMD制御並列処理装置のハードウェア構成を全く変更する必要がない。

【0011】また、第2、第4の観点に係るSIMD制御並列処理方法および装置では、同一グループ内の各要素プロセッサ全てに同一データを入力し、かつ、異なるパラメータを入力することで、データに対しパラメータの異なる処理が完全に並列で実行される。処理結果を各要素プロセッサの認識コードを基に統合処理を行う際に、基準となる要素プロセッサのみ正常の統合処理ができる。また、最終的に、各要素プロセッサの出力を認識コードで選択制御する。これにより、各々のグループで1つ正しい処理結果が出力データとして得られる。この処理では、各グループを仮想的な1つの要素プロセッサと見なせば、その仮想要素プロセッサは個々の要素プロセッサよりも高い処理能力を持つことになる。実質的には、この処理手法によって有効な出力が得られる要素プロセッサ数は減る。ところが、画像処理の分野などでは、一度に並列処理させたい画素は間欠的な場合も多いので、本手法によって、従来は数回に1度しか処理に寄与していなかった要素プロセッサの処理能力を有効活用したことになる。しかも、本実施形態では、この並列処理数を増加させた高効率なSIMD制御プロセッサは、ハードウェアの構成を全く変更する必要がない。

【0012】

【発明の実施の形態】本発明は、たとえば、画像の1水平スキャンラインを1つのデータ・ストリームの単位として、水平方向の画素データを各要素プロセッサPEに1対1に対応させ、次のスキャンラインまでの1水平走査期間中に処理を行う、SIMD制御の多並列画像DSP(Digital Signal Processor)に対して有効なSIMD制御手法を提供する。本発明の実施例として、図とフローチャートを利用し、具体的な事例に基づいて説明していく。

【0013】第1実施形態

第1実施形態では、多並列のSIMD制御プロセッサにおいて、複数の要素プロセッサに対して1つのデータを割り当て、そのデータ処理過程の中間結果等を複数の要素プロセッサに分散して保存することで、実質的に1つのデータ当たりのローカルメモリLM(Local Memory)を複数倍に増大させる手法を提案する。

【0014】まず、本件で前提とするSIMD(Single Instruction Multiple Data)制御の多並列デジタル・プロセッサについて説明する。図1に、実施形態に係るSIMD制御の多並列プロセッサの要部構成を示す。

【0015】このプロセッサ1の内部に多数の要素プロセッサ(Processor Element)が1次元的に配置されている。図1ではその一部として、連続した3つの要素プロセッサPE0、PE1、PE2のみ示している。各要素プロセッサPE0、PE1、PE2の内部には、ローカルメモリ(Local Memory)LM0、LM1、LM2とロジック回路が含まれている。また、これら要素プロセッサ群をSIMD制御するSIMD制御回路10を有する。

【0016】ロジック回路および配線部分の要部は、図1では代表して中央の要素プロセッサPE1についてのみ示している。本実施形態に係るSIMD制御多並列プロセッサ1は、プロセッサとして最低限必要なローカルメモリLM0、LM1、LM2と、ALU(Arithmetic Logic Unit)の他、幾つかの機能を有するロジック回路もしくは配線を必要とする。

【0017】ある要素プロセッサPE1と、その隣接する要素プロセッサPE0、PE2のローカルメモリLM0、LM2とのデータの通信機能を実現するデータ・パス3は、図1のようにそれぞれの要素プロセッサPE0、PE1、PE2について最低左右1本ずつあればよい。これによって、要素プロセッサPE1のALUは、隣接する要素プロセッサPE0、PE2のローカルメモリLM0、LM2のデータを読み込んで処理することができる。また、逆に、要素プロセッサPE1のALUが処理し、ローカルメモリLM1に格納されていたデータを隣接する要素プロセッサPE0、PE2が受信して、自らのデータとして処理することもできる。

【0018】なお、ここでは、以下の説明を簡略化するために、隣接する要素プロセッサPE0、PE2のローカルメモリLM0、LM2とのデータの通信機能を実現するデータ・パス3は隣接左右の要素プロセッサLM0、LM2に対してそれぞれ1本ずつ存在するものとする。もちろん、片側に隣接する2以上の要素プロセッサとのデータ・パスを有する構成でもよいし、これが煩雑であれば、たとえばALU間通信機能を設け、この機能を利用して遠く離れた要素プロセッサとのデータのやり取りも可能である。

【0019】また、要素プロセッサPE1は、そのローカルメモリLM1の記憶値(具体的には、後述するID番号)に応じて、データ・パス3により接続されたローカルメモリLM0、LM1、LM2のいずれかを選択するメモリ選択手段4と、ALUから送られてきた出力データの有効または無効を判定する有効性判定手段5も有する。なお、本実施形態におけるメモリ選択手段4および有効性判定手段5は、実際に物理的構成を新たに付加したものではなく、既存の構成を利用したSIMD制御回路10のプログラムにより実現されている。メモリ選択手段4は、たとえば、メモリ選択用の1ビットのフラグという形で、図示しない制御線の論理を1または0に変化させ、これによりローカルメモリLM0、LM1、LM2に読み出し等の許可を与えることで実現される。また、有効性判定手段5は、要素プロセッサPE0、PE1、PE2の出力データを処理する外部のブロック、たとえば図示を省略した出力回路の既存の機能を利用して、送られてくる出力データを取り込むか否かの制御により実現される。

【0020】このように、本発明が適用できるSIMD制御の多並列プロセッサ1は、通常のALU以外に、各要素プロセッサ間のデータ通信機能、とくに、このデータ通信機能を利用して要素プロセッサの処理に用いるデータを他の要素プロセッサのローカルメモリから読み出す機能と、要素プロセッサの出力の取捨選択を制御する機能とを有することが前提となる。

【0021】図2は、以上のような構成のSIMD制御の多並列プロセッサ1の具体的処理の例を示すフローチャートである。

【0022】まず、ステップST0において、連続した複数の要素プロセッサPE0、PE1、PE2を1つの仮想的なグループ2とするために、要素プロセッサ間のデータ通信機能を利用して、各要素プロセッサ・グループ2、…内のローカルメモリLM0、LM1、LM2にそれぞれ固有のID番号を設定する。このID番号設定では、各要素プロセッサ同士をつなぐデータ・バス3を介して、例えば図示した3つの要素プロセッサPE0、PE1、PE2ごとのID番号ID₀、ID₁、ID₂として0、1、2を付与する。その際、このID番号の配列を0、1、2、0、1、2、0、1、…と他のグループにも同じように繰り返し付与する。そして、各グループ内で固有のID番号を、すべての要素プロセッサ内のローカルメモリの同一アドレスに記憶させる。

【0023】より具体的なID番号設定方法としては、まず、図1で示した3つの要素プロセッサPE0、PE1、PE2を含めたすべての要素プロセッサのうちで1番左に位置するものが、更に左側の要素プロセッサから読み出しを行おうとした場合に、常に読み出し可能なハードウェアの状態にする。たとえば、処理に応じて要素プロセッサがn個のグループにすることが最適であると判断されるが、要素プロセッサの個数をnで割ると端数が生じる場合には、その左端および/または右端の幾つかの要素プロセッサにアクセス禁止を意味するID番号を付与する。これによって、あるプロセッサが処理する際に読み出す要素プロセッサが存在しないということが防止される。そして、まず、すべての要素プロセッサについて、ID番号を格納するアドレスのローカルメモリの値をクリアしておく。

【0024】続いて、アクセスが禁止されていない処理可能な要素プロセッサ群のうち、一番左端の要素プロセッサのID番号を格納するアドレスのメモリ領域にID=0を書き込む。そして、SIMD制御の特質を利用して、他の全ての要素プロセッサに、“左隣りの要素プロセッサPEのIDに1足した値を自らのIDとし、IDの値が3になったら0に戻す”という処理の命令を一括して送る。その結果、最初のID書き込み(ID=0)に続いて、左側から順次、ID番号が1、2、0、1、2、0、1、…と確定してゆく。このようなID番号設定では、要素プロセッサの数が多くても、一番右側の要

素プロセッサまで規則的なID番号が伝搬され、しかも最後のID番号が必ず2となる。確定したID番号は、各ローカルメモリ内の所定アドレスに保存される。

【0025】次に、ステップST1において、入力データを各グループ内の特定の1つの要素プロセッサにのみ配信する。すなわち、たとえば図1に示すID番号ID₀、ID₁、ID₂が左から順に0、1、2となるような3つの連続した要素プロセッサPE0、PE1、PE2を1つの要素プロセッサ・グループ2として、1つの要素プロセッサ・グループ2内のある特定の要素プロセッサに対してのみデータを入力していく。ここでは、3つの要素プロセッサPE0、PE1、PE2の中央の要素プロセッサPE1にのみ有効なデータを入力することとする。

【0026】つぎに、ステップST2の演算処理1に移るが、ここでは、まず、SIMD制御により要素プロセッサPE0、PE1、PE2を含むすべての要素プロセッサに対し、共通した単一の処理命令を発行する。ただし、前記したように実際に有効なデータが入力されているのは中央の要素プロセッサPE1のみである。そして、与えられた処理命令に沿って、各要素プロセッサPE0、PE1、PE2が演算処理1を実行する。

【0027】いま、この処理の途中で、中央の要素プロセッサPE1は、演算過程でその中間処理結果をテンポラリ・データとして自らのローカルメモリLM1に保存しておきたいが、これを保存すると、そのメモリ容量が限界近くに達してしまう場合がある。このような場合、実際に有効なデータを処理していない左右の要素プロセッサPE0、PE2のローカルメモリLM0、LM2のメモリ領域を一時的なデータ保存用に利用する。また、自らのローカルメモリLM1に十分な空き領域がある場合でも、後の合成処理時のデータの一時退避用などの用途に、この空き領域はとっておきたいこともあり、この場合も、ローカルメモリLM0、LM2を一時的に利用する。いずれにしても処理途中の演算結果を中間結果1として出力するのであるが、ここでの中間結果1（後述する中間結果2も同様）は、それに続く処理に直ぐには用いられないような演算結果を想定している。たとえば繰り返し計算の幾つかのルーチンワークが完結した時点の演算結果などが、ここで言う中間結果に該当する。ただし、出力したい演算結果が、続く処理で用いるような演算結果とならざるを得ない場合も多い。その場合、以下の説明では特に言及しないが、つぎのステップST3、ST4等を経て保存した後、通常の通信機能があるSIMD制御プロセッサが行っているように、その時々で適宜、ローカルメモリLM0またはLM2に保存している中間結果を読み出して処理に利用することになる。

【0028】中間結果1の保存に先立って、まず、ステップST3において、要素プロセッサPE0、PE1、PE2それぞれにおいて、自己のID番号が、有効デー

タを入力した要素プロセッサPE1以外の所定のID番号と一致するかが判断される。この例では、まず、所定のID番号($ID_0 = 0$)と自己のID番号との一致が判断される。着目している中央の要素プロセッサPE1と、その右隣の要素プロセッサPE2は、ともにID番号が不一致なので、中間結果1を保持したまま、他の要素プロセッサから中間結果1の掃き出しを要求されるまで待機状態にある。所定期間待って掃き出し要求がない場合は、処理を続行する。

【0029】一方、左側の要素プロセッサPE0は、ID番号が一致するので、ステップST4において、共通の処理命令に従って中間結果1の保存を行う。この処理命令には、たとえば“自己のIDが $ID = m - 1$ (m : 有効データを入力したPEのID番号)ならば、PE m の中間結果1をLM($m - 1$)に保存せよ”といった指示が含まれている。したがって、要素プロセッサPE0は、隣の要素プロセッサPE1に中間結果1を要求し、送られてきた中間結果1を、自己のローカルメモリLM0に保存した後、次の処理ステップに進む。要素プロセッサPE1は、中間結果1を排出した時点で、次の処理ステップに進む。

【0030】各要素プロセッサPE0, PE1, PE2は、同様にして、ステップST5で演算処理2を実行する。この処理途中に、再びテンポラリ・データを保存する必要が生じれば、上記と同様にして、ID番号の一致判断とデータ保存を行う。すなわち、ステップST6において、自己のID番号と、ステップST3で使用しなかった他のID番号($ID = 2$)との一致が判断される。着目している中央の要素プロセッサPE1と、その左隣の要素プロセッサPE0は、ともにID番号が不一致なので、中間結果2を保持したまま、他の要素プロセッサから中間結果2の掃き出しを要求されるまで待機状態にある。所定期間待って掃き出し要求がない場合は、処理を続行する。

【0031】一方、右側の要素プロセッサPE2は、ID番号が一致するので、ステップST7において、共通の処理命令に従って中間結果2の保存を行う。この処理命令には、たとえば“自己のIDが $ID = m + 1$ (m : 有効データを入力したPEのID番号)ならば、PE m の中間結果2をLM($m + 1$)に保存せよ”といった指示が含まれている。したがって、要素プロセッサPE2は、隣の要素プロセッサPE1に中間結果2を要求し、送られてきた中間結果2を、自己のローカルメモリLM2に保存した後、次の処理ステップに進む。要素プロセッサPE1は、中間結果2を排出した時点で、次の処理ステップに進む。

【0032】この時点で、あるPEグループ2の3つの要素プロセッサPE0, PE1, PE2の中央の要素プロセッサPE1から見て、左右の要素プロセッサPE0, PE2のローカルメモリLM0, LM2には、それ

ぞれ自分が行った演算処理の中間結果1, 2が格納されている。したがって、要素プロセッサPE1にとって、ローカルメモリLM0, LM2を、あたかも自己のローカルメモリLM1の一部として機能させたかのような状態になる。一方、他の2つの要素プロセッサPE0, PE2においても、同様に中間結果1, 2は算出されるが、もともと有効なデータが入力されていないうえ、このような大きなメモリ空間が必要な演算処理は正しく実行できない。中間結果1, 2を保管する場所がないからである。

【0033】以上より、1つのPEグループ2を仮想的な1つの要素プロセッサPEと見なした場合、元の要素プロセッサの3倍のローカルメモリ容量を持てることが分かる。

【0034】つぎのステップST8において、要素プロセッサPE0, PE1, PE2は、最終の演算処理3を行う。この演算処理3の結果が、そのまま最終の出力データとなるような場合は、ステップST9およびST10をスキップして、処理フローがステップST11に進む。

【0035】演算処理3の結果を先に算出しておいた中間結果1, 2の何れかと合成したい場合は、次のステップST9において合成処理1が実行される。このとき中間結果1を読み出して合成処理1に用いる場合は、(自己のID番号-1)のID番号の要素プロセッサから中間結果1を読み出す。また、中間結果2を読み出して合成処理1に用いる場合は、(自己のID番号+1)のID番号の要素プロセッサから中間結果2を読み出す。このような処理命令はすべての要素プロセッサPE0, PE1, PE2に共通である。このため、有効データを用いた要素プロセッサPE1以外の他の要素プロセッサPE0, PE2は、正常な中間結果の読み出しができない。たとえば要素プロセッサPE2にとっては、前記したように中間結果を両隣の要素プロセッサに保存できず、かりに自己のメモリ容量に余裕がある場合は、そこに中間結果1, 2が保存されているはずである。にもかかわらず、要素プロセッサPE2は、この共通の命令により隣の要素プロセッサPE1(または、存在していればPE4)から全く無関係の中間結果を誤って読み出してしまふ。したがって、合成処理1後の結果は、全く意図しない結果となってしまふ。このことは、もう1つの要素プロセッサPE0においても同じである。

【0036】このようにSIMD制御で共通の命令が発行されることを上手に利用することで、有効データが与えられた中央の要素プロセッサPE1に限って、正しいデータの読み出しができ、したがって正常に合成処理1が実行できることが保証される。この合成処理1の結果を最終的な出力データとする場合は、処理フローが次のステップST10をスキップする。

【0037】さらに、合成処理1の結果を未だ合成して

いない他の中間結果と合成したい場合は、つぎのステップST10において合成処理2が実行される。このときのデータ読み出しも、上記合成処理1のときと同じように、中央の要素プロセッサPE1のみが正しくでき、他の要素プロセッサPE0、PE2では誤読み出しになってしまう。したがって、中央の要素プロセッサPE1だけから、正しい合成処理2の結果が出力される。

【0038】つぎに、ステップST11～ST13で出力データの有効性判定を行う。上記したように、中央の要素プロセッサPE1のみ正しい演算結果（あるいは合成結果）が得られているのであるから、一見、有効性判定の判断は必要ないように思われる。しかし、当該SIMD制御多並列プロセッサから外部にデータ出力するときは正しいものだけが出力されるのであるから、そのためには何らかの出力選択が行われる。いわゆる出力ポイント回路の動作、すなわち出力部が有効性判定フラグを用意して、それを1（有効）または0（無効）に変化させるなどといった制御に限らず、たとえば出力部で要素プロセッサPE1以外の結果は無視するといったことも、広い意味で、この有効性判定に含まれる。また、必ず2つ以上のローカルメモリを利用しなければ処理ができない場合にあっては、共通の命令に“自己のメモリが限界に達したときは処理を停止する”ことを含ませておけば、中央の要素プロセッサPE1以外からは出力データそのものが送られてこない。このような制御も、ここでいう有効性判定の1種である。

【0039】図2の例では、ステップST11において、PEグループ2内の全てのID番号が調べられ、ID=1である場合のみ、その要素プロセッサPE1の出力データをステップST12で有効なものとして、ステップST13で出力される。一方、ID=1でない他の要素プロセッサPE0、PE2の出力データはステップST14で無効であるとされてステップST13で出力されないか、あるいは、ステップST14で無効処理された後に出力される。この無効処理された出力データは、外部には出力されない。

【0040】本実施例におけるデータの流れを分かりやすく表すと、図3のようになる。なお、上記説明では、中央の要素プロセッサPE1に有効データを入力した場合を説明したが、当然、両端の要素プロセッサPE0、PE2に対して有効データが入力される場合もある。この場合、中間結果を保存する要素プロセッサが存在しないという事態が想定されるが、これは、処理の最初にステップST0で、有効データが入力される要素プロセッサが必ず中央に位置するようにPEグループ分けを変更することで対処する。あるいは、ID番号0、1、2は環状的に変化する規則にしておけば、この事態は避けられる。たとえば、有効データを入力した要素プロセッサが左端のPE0であるとき、中間結果1の保存で、前記した“自己のIDがID=m-1（m：有効データを入

力したPEのID番号）ならば、PEmの中間結果1をLM（m-1）に保存せよ”との指令にしたがうと、自己のID=-1となって、そのようなID番号は存在せず、中間結果1は保存されない事態となる。この事態を避けるためID番号を環状的に変化させると、ID=-1=2であるから、要素プロセッサPE2が中間結果1を保存する。そして、読み出し時の命令に含まれる（自己のID番号-1）=-1=2となって、正常に中間結果1を読み出すことができる。この方法では、PEグループ内で図2の処理フローを繰り返す場合、2回目以降はステップST0を省略できるという利点がある。

【0041】また、上記説明では3つの要素プロセッサPE0、PE1、PE2を1つのPEグループ2として扱ったが、隣接する要素プロセッサ間のローカルメモリに対するデータ通信機能を利用すれば、理論的にPEグループ2内の要素プロセッサの数に上限はない。このとき、より離れた要素プロセッサのローカルメモリのデータを参照する場合、中間に位置する要素プロセッサは単に中継させるだけにしてバケツリレー形式でデータ転送すれば、ハードウェアとしてのデータ・パス3の追加は必要ない。

【0042】第1実施形態では、複数の要素プロセッサPE0、PE1、PE2のローカルメモリLM0、LM1、LM2を1つのデータに対して仮想的に割り当てて使用する。このため、PEブロック2内の要素プロセッサ数をnとした場合、SIMD制御プロセッサ1として一度に処理できるデータ数は1/nになるものの、1つのデータに対して使用できるローカルメモリを実質的にはn倍にすることが可能となる。このような仮想的なメモリ拡張手法は、制御部のプログラムで行えるため、従来のSIMD制御プロセッサのハードウェア構成を全く変更することなしに実現できる点で極めて有用である。

【0043】第2実施形態

第2実施形態では、複数の要素プロセッサそれぞれに1つの有効なデータを割り当てるが、同時に、同じデータが割り当てられた要素プロセッサに対して、互いに異なるパラメータを入力し、共通の命令コードにしたがって処理を行うことで、実質的に1つのデータ当たりの処理ステップ数を複数倍に増大させる手法を提案する。

【0044】この処理では、図1と同じ構造のSIMD制御プロセッサを用いる。また、処理の具体例としては、図4に示すように、3つの並列のFIRフィルタリング処理P1と、それらの処理結果をミキシングする処理P2がある。FIRフィルタリング処理P1で、原画像の異なる複数のサンプリング点のデータに所定の係数で重み付けする。そして、次のミキシング処理P2で、重み付けした各データを合成する。この合成後の画像データは、新たなサンプリング点で欠落する画像情報が補間されたものとなる。

【0045】図5は、第2実施形態に係る処理手順を示すフローチャートである。まず、ステップST0において、第1実施形態と同じ手法によって、0, 1, 2, 0, 1, 2, 0, 1, …と各要素プロセッサのローカルメモリにID番号を記憶させ、ID番号ID₀, ID₁, ID₂が左から0, 1, 2と並ぶ3つの要素プロセッサPE0, PE1, PE2を、まとめて1つの要素プロセッサ(PE)グループ2とする。

【0046】ステップST21では、第1実施形態とは異なり、PEグループ2内の各要素プロセッサPE0, PE1, PE2に同一の有効なデータを入力する。ここではFIRフィルタリング処理であるので、データとしては原画像のサンプリング時の画像データ、たとえば輝度Yまたは色相Cr, Cbなどのデータであり、PEグループ内の複数の要素プロセッサでは同一のものが用いられる。この場合、必ずしも各要素プロセッサPE0, PE1, PE2すべてに有効なデータを入力する必要はなく、PEグループ2内の、たとえば中央の要素プロセッサPE1にのみデータを入力し、それをグループ2内の他の要素プロセッサPE0, PE2に内部で配信してもよい。

【0047】続くステップST22において、それら各要素プロセッサPE0, PE1, PE2に、データとは別に、複数のFIRフィルタのフィルタ係数を、互いに相異なる組み合わせパターンのパラメータ・セットとして入力する。パラメータ・セットは、含まれているパラメータ(フィルタ係数)の組合せが3つとも全く異なるが、少なくとも中央の要素プロセッサPE1の出力データ生成に必要なパラメータが分散して含まれる必要がある。

【0048】そして、ステップST23において、入力したデータおよびパラメータ・セットを用いてFIRフィルタリング処理P1を実行する。この処理時に、SIMD制御プログラムの共通命令コードが発行され、これに従ってすべての要素プロセッサPE0, PE1, PE2が同一手順で並列にFIRフィルタリング処理P1を演算処理する。

【0049】FIRフィルタリング処理P1後は、つぎのステップST24およびST25において、ミキシング処理P2を実行する。具体的に、たとえば図5に示すように、最初のステップST24で合成処理1が行われ、その合成結果に対し、さらに次のステップST25で合成処理2が施される。これらの合成処理1, 2は、たとえば図1で図示を省略したALU間通信パスを利用して、それぞれの要素プロセッサPE0, PE1, PE2が、自分を中心として左右の要素プロセッサからの処理結果を取り込みながら行う。

【0050】たとえば、最初の合成処理1(ステップST24)の共通コマンドが、“自己の処理結果を(自己のID番号)-1のPEの処理結果と合成せよ”であ

り、次の合成処理2(ステップST25)共通コマンドが、“自己の合成処理1の結果を(自己のID番号)+1のPEの処理結果と合成せよ”であるとする。このコマンドにしたがうと、要素プロセッサPE1は要素プロセッサPE0の処理結果を用いて合成処理1を行い、その結果をさらに要素プロセッサPE2の処理結果と合成し、出力データとして出力する。ところが、要素プロセッサPE0は合成処理1時に、自己の処理結果を、当該PEグループ2の左隣の他のPEグループでの処理結果と合成してしまい、誤った処理を行ってしまう。同様に、要素プロセッサPE2は合成処理2時に、自己の合成処理1の結果を、当該PEグループ2の右隣の他のPEグループでの処理結果と合成してしまい、誤った処理を行ってしまう。したがって、正しい出力データが得られるのは、中央の要素プロセッサPE1のみである。

【0051】その後は、第1実施形態と同様の手順により、ステップST11~ST13で出力データの有効性判定を行う。この最後のステップST13では、たとえば中央の要素プロセッサPE1のみから結果が出力され、図5で示した処理フローが完了する。この処理手法では、3つの要素プロセッサPE0, PE1, PE2を1つのPEグループ2とし、仮想的な1つの要素プロセッサと見なせば、通常の要素プロセッサの3倍の処理能力で、1つのFIRフィルタリング処理を行うことになる。

【0052】なお、第2実施形態では、ID番号を環状的に変化させることはなく、また、1つのPEグループ2内の要素プロセッサ数に理論的に上限はない。

【0053】このように、第2実施形態では、同一PEグループ内の各要素プロセッサ全てに有効データを入力し、かつ、PEグループ内の各要素プロセッサに対して異なるパラメータ・セットを入力すれば、各有効データに対しパラメータの異なる処理を完全に並列で実行できる。処理したデータを各要素プロセッサのID番号で区別して統合処理を行ったうえで、最終的に、各要素プロセッサの出力を、たとえばID番号で選択制御する。これにより、各々のPEグループで1つ正しい処理結果が出力データとして得られる。この処理では、各グループを仮想的な1つの要素プロセッサと見なせば、その仮想要素プロセッサは個々の要素プロセッサよりも高い処理能力を持つことになる。実質的には、この処理手法によって有効な出力が得られる要素プロセッサ数は減る。ところが、画像処理の分野などでは、一度に並列処理させたい画素は間欠的な場合も多いので、本手法によって、従来は数回の処理に1度しか処理に寄与していなかった要素プロセッサの処理能力を有効活用したことになる。しかも、本実施形態では、この並列処理数を増加させた高効率なSIMD制御プロセッサを、ハードウェアの構成を全く変更することなしに実現できた。

【0054】

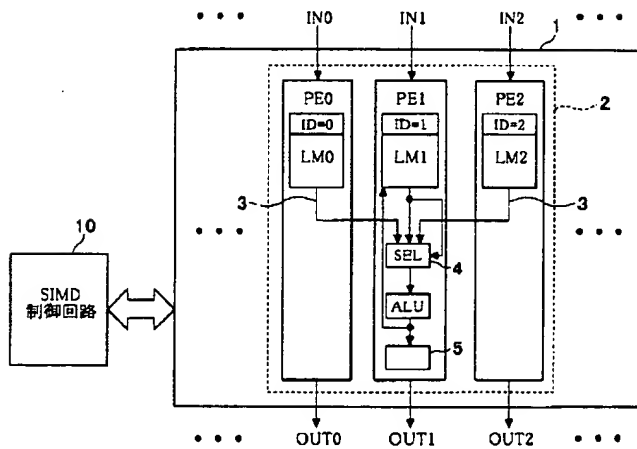
【発明の効果】本発明に係るSIMD制御並列処理方法および装置によれば、要素プロセッサが無駄になるデータ・ストリームに対して、要素プロセッサを有効利用したうえでローカルメモリを実質的に複数倍に増加させる処理を、ハードウェアの構成をそのままに実現することを可能とした。

【0055】また、本発明に係る他のSIMD制御並列処理方法および装置によれば、FIRフィルタリング処理など、パラメータ（フィルタ係数）が異なるだけで全く同じような処理を繰り返す並列処理数を、ハードウェアの構成をそのまま増加させて、処理効率を上げることを可能とした。

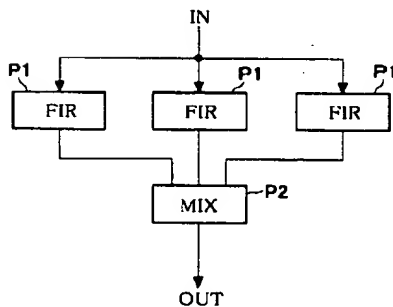
【図面の簡単な説明】

【図1】本発明の実施形態に係るSIMD制御の多並列デジタル・プロセッサの要部構成を示すブロック図である。

【図1】



【図4】



【図2】第1実施形態の処理手順を示すフローチャートである。

【図3】第1実施形態における、各要素プロセッサ間のデータの流れを示す模式図である。

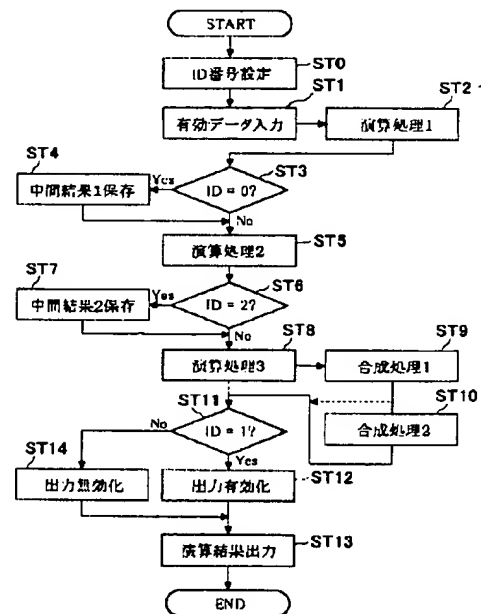
【図4】第2実施形態で実施する処理ブロックを示す図である。

【図5】第2実施形態の処理手順を示すフローチャートである。

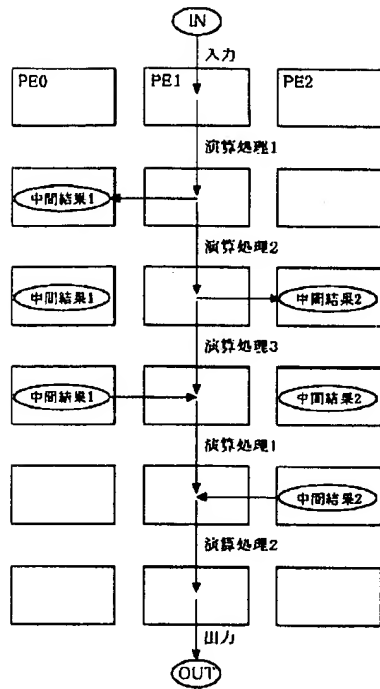
【符号の説明】

1…SIMD制御の多並列デジタル・プロセッサ（SIMD制御並列処理装置）、2…仮想的な要素プロセッサ・グループ、3…データ・パス、4…メモリ選択手段、5…有効性判定手段、PE0、PE1、PE2…要素プロセッサ、LM0、LM1、LM2…ローカルメモリ（メモリ部）、ALU…処理部、ID₀等…ID番号（認識コード）。

【図2】



【図3】



【図5】

